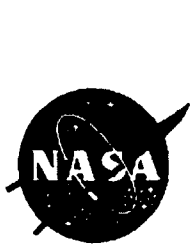
System Risk Balancing Profiles: Software Component

John C. Kelly & Burton C. Sigal
Quality Assurance Office

Tom Gindorf
Assurance Technology Program Office

NASA Jet Propulsion Laboratory,
Pasadena, CA

Office of Engineering and Mission Assurance, Assurance Technology Programs



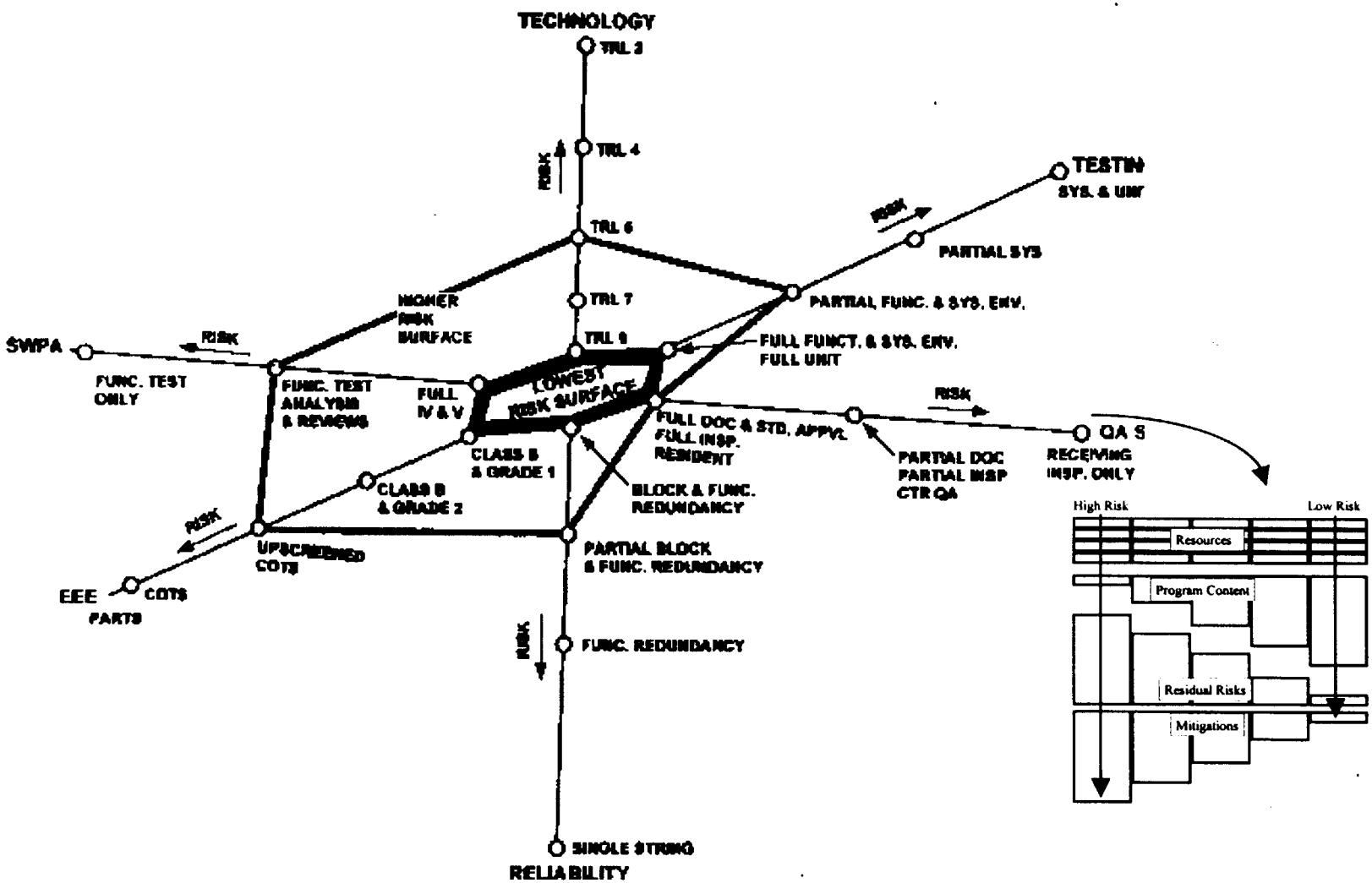
Background*

JPL

- NASA's new environment:
  - From Few to Many Projects
  - From Large to Small Projects
  - From Single Monumental Success to Many Opportunities for Success
  - From a Large Budgets to Declining or Flat Budgets
  - From Conservative Risk Avoidance to Risk Management based on cost of failure
- Risk is a resource that can be traded like other resources (mass, power, performance, schedule, & cost)
- For the reasonably small number of critical subsystems which could affect human life, the traditional risk avoidance approach should still be used

*Based on Greenfield & Gindorf, "Risk as a Resource - A New Paradigm", 1996

What can be done to balance the wide variety of risks a project needs to manage?

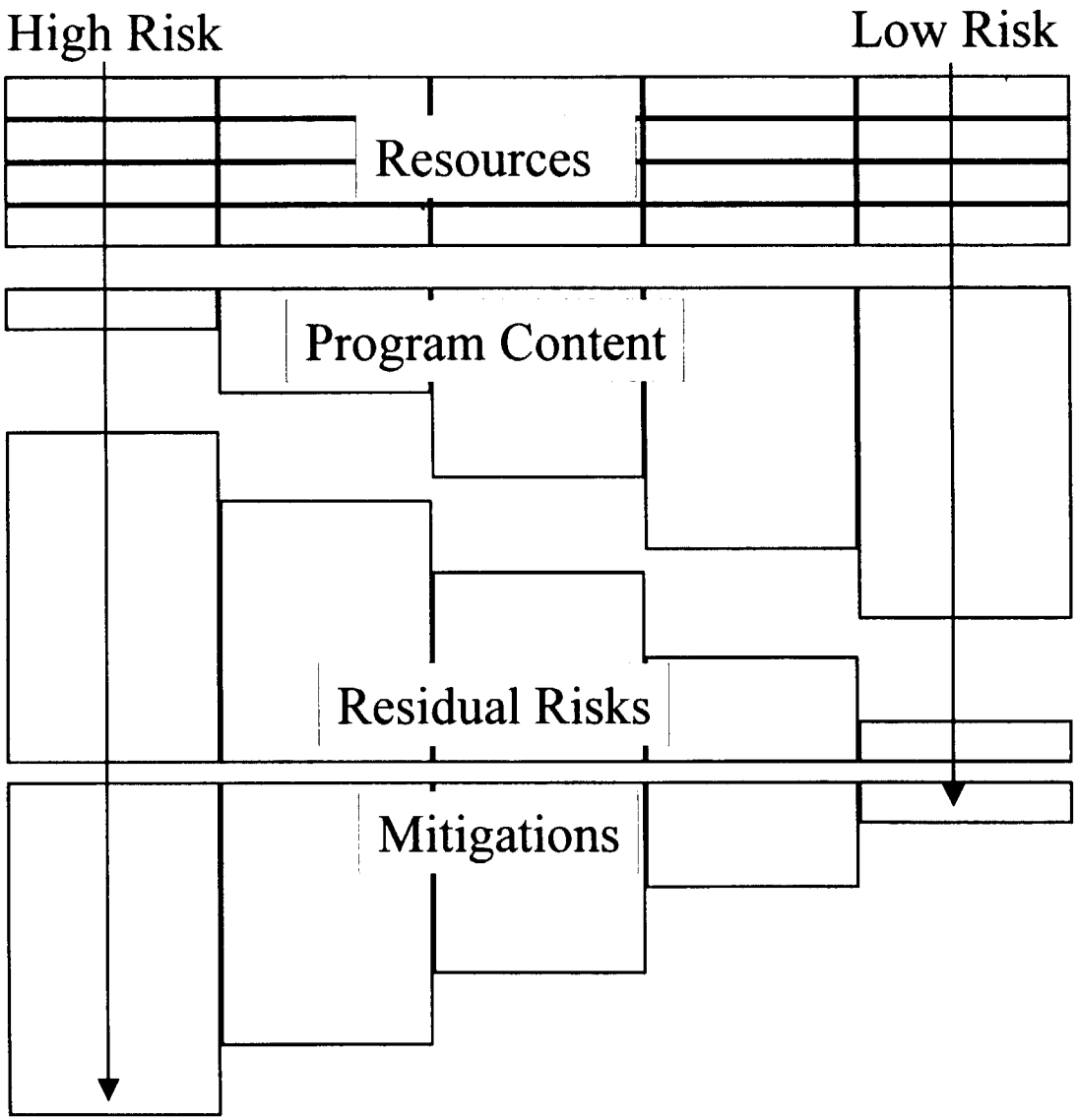


JPL

Approach

- **Provide a mechanism for identifying performance risk associated with program content**
- **Identify mitigation possibilities corresponding to residual performance risk**
- **The full set of charts will address balancing risk involving System, Hardware and Software**
- **Risk associated with people, organizations, and facilities are not within the scope of this work, but need to be addressed separately**

Overview:
"FODORs Charts"



Resources*

JPL

- Drivers
  - Spacecraft & Science Performance
  - Cost
  - Schedule
- Planning Cautions
  - Cost Risk Factors
  - Schedule Risk Factors
- Mass
- Power

* Note: Resources include some areas which may not be applicable to software (i.e. mass & power)

Software Quality Assurance / V&V Program Content

JPL

- **Component Areas**
  - Testing
  - Analysis
  - Quality Assurance
  - Related Management
  - Other
- **Used common "tried and proven" software QA / V&V activities to populate the program content**
- **Used a history of QA / V&V services that were previously provided for JPL and NASA projects (on project funds)**
- **Advanced approaches, software QA / V&V research, and "piloted only" techniques were not included in the program content area**
- **Content ranged from a super minimal approach to a full up QA / V&V program**

Software Quality Assurance / V&V Program Content (continued)

JPL

- **Included software safety and hazards analysis as required by NASA policy even in the minimal program category**
- **SEI's Capability Maturity Model was originally used to segment Software Quality and V&V into consistent levels**
- **Qualitative differences in the individual program content activities were noted**
  - Acceptance Test (pass/fail)
  - Acceptance Test (w/metrics & key critical functions)
  - Acceptance Test (w/metrics, good functional coverage, & witnessing)
  - Acceptance Test (w/metrics, full functional coverage, & witnessing)

Residual Performance Risks

JPL

- **"What are the risks, if projects chose not to do individual program content items"?**
  - We went through the program content list asking ourselves:
    - "If this QA /V&V activity is deleted, what can/has go wrong? and
    - "If this QA /V&V activity is used correctly, what problems/risks should be avoidable?
- **In today's NASA environment, the full up "Low Risk" QA / V&V program can only be justified for a few isolated projects**
- **Raised excellent questions regarding the content area from a project management viewpoint:**
    - If I don't choose or have funds to have particular QA / V&V program content areas, what risks are being accepted by the project?
    - Are there redundancies in program content items with respect to individual risks?
    - Are there risks that have insufficient coverage by standard QA / V&V program content areas?
    - Given a limited budget and specific project resource drivers for QA / V&V, is the project buying the best program content?

Mitigations

JPL

- Project factors or techniques which reduce/eliminate the risks associated with software
- Given a set of typical program content items, risks can be reduced by:
  - Adding program content from another column when considering the aggregate of risks
  - Utilizing mitigations from the same column
- Mitigations include
  - Advanced techniques (Formal Methods, Model Checking, Simulation, etc.)
  - Opportunistic factors (reusing high quality software components, etc.)
- Mitigations need to be carefully selected with assistance from someone with expertise in a broad spectrum of software QA / V&V techniques

The Early Mapping of SEI's Capability Maturity Model into the FODOR Chart

SEI Level 1 KPAs	SEI Level 2 KPAs	Tailored Approach	SEI Level 2 & 3 KPAs	SEI Level 2, 3, 4 & 5 KPAs
	• Configuration Management (K1) • Software Quality Assurance (K2) • S/W Subcontract Management (K3) • S/W Project Tracking and Oversight (K4) • S/W Project Planning (K5) • Requirements Management (K6)		• Configuration Management (K1) • Software Quality Assurance (K2) • S/W Subcontract Management (K3) • S/W Project Tracking and Oversight (K4) • S/W Project Planning (K5) • Requirements Management (K6) • Peer Review (K7) • Intergroup Coordination (K8) • S/W Product Engineering (K9) • Integrated S/W Management (K10) • Training Program (K11) • Org. Process Definition (K12) • Org. Process Focus (K13)	• Configuration Management (K1) • Software Quality Assurance (K2) • S/W Subcontract Management (K3) • S/W Project Tracking and Oversight (K4) • S/W Project Planning (K5) • Requirements Management (K6) • Peer Review (K7) • Intergroup Coordination (K8) • S/W Product Engineering (K9) • Integrated S/W Management (K10) • Training Program (K11) • Org. Process Definition (K12) • Org. Process Focus (K13) • Quality Management (K14) • Process Measurement and Analysis (K15) • Process Change Management (K16) • Technical Innovation (K17) • Defect Prevention (K18)

Early version of Guide

JPL

- <insert version 20 of the guide here, see attached file "A_Balance.doc">

Evolution of the Guide

JPL

- Reduced the number of columns from 5 to 3 to show only the extremes
- Introduced identifiers to trace
  - Missing or Weak Program Content to Residual Risks
  - Mitigations to Residual Risk

Software QA / V&V Guide

JPL

- <insert the latest guide here, see the attached file "B_Balance.doc">

"Reading" the Guide

JPL

- The guide is a starting place to tailor a project/mission in a number of support domains. The center column is for the "results" of the tailoring decisions and could be a basis for the Software QA or V&V Plan.
- The left columns represent a High Risk (low Software QA / V&V content) while the right hand column represents a Low Risk (high Software QA / V&V content) approach.
- The top "group" of each column (Program) contains the software related activities, divided into five (5) areas, Testing,Analysis, QA, (Related) Management, and Other. Each element has a reference designator for tractability purposes.
- The center group (Residual Performance Risks) contains the residual risks that occur because of the activities that are NOT to be done by the program.

"Reading" the Guide (continued)

JPL

- After identifying the residual risks, there are choices. The programs can:
  - Use some/all of the mitigation strategies in the lower group (Mitigations) to reduce/mitigate the associated risk,
  - Change their minds and do the (upper group) activity to eliminate risks,
  - Do both,
  - Decide to do nothing and accept the risk.
- The numbers after each mitigation strategy trace to the residual risk it is intended to mitigate. Notice that several strategies address the same risk, and many strategies address multiple risks, so the projects will have a number of cost-benefit tradeoffs they can make in managing risk.
- There are no 100% certain, 0% Risk programs!

What the Guide "is" and what it "is not"

JPL

- The Guide is:
  - Useful for identifying project risk associated with a level of QA /V&V program content
  - Identifying mitigation possibilities
  - Helpful in planning appropriate resources for QA / V&V program content (and balancing resources across various project risk reduction areas)
- The Guide is not:
  - a substitute for an experts' participation during the planning process
  - prescriptive in nature (it is intended to illustrate how to tailor a QA / V&V program)
  - a process monitoring and corrective action technique (needed by projects beyond the use of this guide)

Summary

JPL

- The Software QA / V&V guide will be reviewed and updated based on feedback from NASA organizations and others with a vested interest in this area
- Hardware, EEE Parts, Reliability, and Systems Safety are a sample of the future guides that will be developed
- Cost Estimates, Lessons Learned, Probability of Failure and PACTS (Prevention, Avoidance, Control or Test) are needed to provide a more complete risk management strategy
- This approach to risk management is designed to help balance the resources and program content for risk reduction for NASA's changing environment

Risk Balance Profile
Software Quality and V&V Program Guide
"FODORS"

Performance	<----------	Based on Trade-offs of Risk,	Mitigation, Content	determined by user	---------->
Costs	<----------	Based on Trade-offs of Risk,	Mitigation, Content	determined by user	---------->
Schedule	<----------	Based on Trade-offs of Risk,	Mitigation, Content	determined by user	---------->
COST RISK FACTORS	⟨ Schedule Pressure Resolved by $ ⟨ Repeat Testing ⟨ Changing Requirements ⟨ S/W Faults Could Impact System Testing	⟨ Schedule Pressure Resolved by $ ⟨ Repeat Testing ⟨ Changing Requirements ⟨ S/W Faults Could Impact System Testing/Schedule	***Determine***	⟨ Schedule Pressure Resolved by $ ⟨ Repeat Testing ⟨ Changing Requirements ⟨ S/W Faults Could Impact System Testing	*(TBD)*
SCHEDULE RISK FACTORS	⟨ Late Problem Identification ⟨ Repair and Repeat Testing ⟨ Changing Requirements ⟨ S/W Faults Could Impact System Testing	⟨ Late Problem Identification ⟨ Repair and Repeat Testing ⟨ Changing Requirements ⟨ S/W Faults Could Impact System Testing	***Determine***	⟨ Late Problem Identification ⟨ Repair and Repeat Testing ⟨ Changing Requirements ⟨ S/W Faults Could Impact System Testing	*(TBD)*
Mass	Negligible or small	Negligible or small		Negligible or small	Negligible or small
Power	Negligible or small	Negligible or small		Negligible or small	Negligible or small
Program Title	**Very High Risk Minimal QA/V&V Program**	**Medium/ High Risk**	**Tailored Approach**	**Medium Risk**	**Low Risk Complete QA/V&V Program**
Software Program Contents	**Program Content** **Testing** ⟨ T1-Accept Test (pass/fail w/o metrics) ⟨ T2-Functional Test (pass/fail) **Analysis** ⟨ A1-Hazards Analysis ⟨ A2-S/W FMEA (if applicable for critical functions only) **QA** ⟨ None **Related Management** ⟨ None **Other** ⟨ None	**Program Content** **Testing** ⟨ T1-Accept Test (w/ *Metrics & Key Critical Functions*) ⟨ T2-Functional Test (w/ *Metrics & Key Critical Functions*) ⟨ T3-Subsystem integration Test ⟨ T4-Unit Test (basic SW Dev Folders). ⟨ T5-Formal Test Plan **Analysis** ⟨ A1-Hazards Analysis ⟨ A2-S/W FMEA (critical functions) **QA** ⟨ Q1-Conformance to S/W Standards & Guidelines ⟨ Q2-Requirements Trace ⟨ Q3-Basic Technical Status Reviews (TSRs) including critical design and select code ⟨ Q4-Light V&V role (report to Proj Mgr.) ⟨ Q5-Requirements Mgt. (local config. mgt.) **Related Management** ⟨ M1-Minimal S/W QA Plan (WPA only) ⟨ M2-Configuration Management (Code & version control) ⟨ M3-Milestone Reviews (CDR, PDR, ...) **Other** ⟨ O1-Support Contractor Mgt. (Assessment of critical areas)	**Program Content** *As Selected (Tailored to be Project Specific)*	**Program Content** **Testing** ⟨ T1-Accept Test (w/ Metrics, good functional coverage, & witnessing) ⟨ T2-Full Functional Test (w/ Metrics) ⟨ T3-Subsystem integration Test (Metrics) ⟨ T4-Unit Test (full SW Dev Folders) ⟨ T5-Formal Test Plan **Analysis** ⟨ A1-Hazards Analysis ⟨ A2-S/W FMEA ⟨ A3-Safety Analysis (critical issues) ⟨ A4-Code Analysis (of critical w/automated support) **QA** ⟨ Q1-Conformance to S/W Standards & Guidelines (QA check/peer audit) ⟨ Q2-Requirements Trace ⟨ Q3-Defined Peer Reviews used for TSRs ⟨ Q4-*Reporting to Center Director)* ⟨ Q5-Requirements Mgt. (trace CM, CCB) ⟨ Q6-Operations Software QA & V&V (critical functions updates only) **Related Management** ⟨ M1-Full S/W QA Plan ⟨ M2-Configuration Management (Code & Version control) ⟨ M3-Milestone Reviews (CDR, PDR, etc.) ⟨ M4-Risk Management program (basic) ⟨ M6-Project S/W Metrics program (System/Acc. P/FRs) **Other** ⟨ O1-Support Contractor Mgt. (continuous assessment) ⟨ O2-Mission Operations and Command Assurance (MOCA)	**Program Content** **Testing** ⟨ T1-Accept Test (w/ Metrics, full functional coverage, & witnessing) ⟨ T2-Full Functional Test (w/ Metrics) ⟨ T3-Subsystem integration Test (Metrics / trend analysis) ⟨ T4-Unit Test full SW Dev Folders) ⟨ T5-Formal Test Plan **Analysis** ⟨ A1-Hazards Analysis ⟨ A2-S/W FMEA ⟨ A3-Safety Analysis (Full) ⟨ A4-Code Analysis (Full) ⟨ A5-S/W Fault Tree Analysis **QA** ⟨ Q1-Conformance to S/W Standards & Guidelines (QA critical item audit) ⟨ Q2-Requirements Trace (complete) ⟨ Q3-S/W Inspections(NASA) used for TSRs (*w/ increased coverage*). ⟨ *Q4-IV&V w/ independent reporting to NASA HQ* ⟨ Q5-Requirements Mgt. (trace CM, CCB, tool, *volatility tracking)* ⟨ Q6-Operations Software QA & V&V (incremental updates) **Related Management** ⟨ M1-Full S/W QA Plan ⟨ M2-Configuration Management *(Full coverage w/ mandatory use of a tool)* ⟨ M3-Milestone Reviews (CDR, PDR, etc. *with participation of independent reviewers mandatory)* ⟨ M4-Project Risk Management program ⟨ M5-Integrated Support of Fault Protection and/or Failure Detection, Isolation & Recovery subsystems ⟨ M6-Full Project Software Metrics program **Other**
RESIDUAL PERFORMANCE RISK	**Residual Risks** ⟨ R1- Lack of confidence in acceptability of S/W to meet system's needs-T1 ⟨ R2 - Unknown functional and system margins-T2 ⟨ R3 - Inconsistent S/W requirements with respect to the system's functional requirements (FRD)-Q2 ⟨ R4 - Incorrect design functionality-Q2 ⟨ R5 - No regression testing - T5, M4 ⟨ R6 - S/W builds not converging to an acceptable product – T5, M2 ⟨ R7 - Inputs to S/W could violate boundary conditions, trigger non-tested paths, etc. – T5, Q2 ⟨ R8 - Poor Workmanship in the software product (spaghetti code, un-maintainable code, etc.) – Q1, Q3 ⟨ R9 - Latent S/W defects could cause the system to fail or not meet it's requirements- T5, Q2, Q5 ⟨ R10 - Late awareness (or lack of anticipation) of schedule, performance, cost and quality problems – T5, Q5, M2, M3 ⟨ R11 - Software safety problem – A2, A3 ⟨ R12 - Executing faulty commands on a spacecraft – Q1, O2 ⟨ R13 - Lack of robustness of functions supported by S/W – Q3, Q5, A4 ⟨ R14 - S/W fails in a harmful manner – A1, A2	**Residual Risks** ⟨ R1- Lack of confidence in acceptability of S/W to meet system's needs-T1 ⟨ R7 - Inputs to S/W could violate boundary conditions, trigger non-tested paths, etc. – T5, Q2. ⟨ R8 - Poor Workmanship in the software product (spaghetti code, un-maintainable code, etc.) – Q1, Q3 ⟨ R9 - Latent S/W defects could cause the system to fail or not meet it's requirements- T5, Q2, Q5 ⟨ R10 - Late awareness (or lack of anticipation) of	**Residual Risks** ***Appropriate Subset of Residual Risk Issue Relating to Selected Program Content***	**Residual Risks** ⟨ R7 - Inputs to S/W could violate boundary conditions, trigger non-tested paths, etc. – T5, Q2. ⟨ R9 - Latent S/W defects could cause the system to fail or not meet it's requirements- T5, Q2, Q5 ⟨ R11 - Software safety problem – A2, A3 ⟨ R12 - Executing faulty commands on a spacecraft – Q1, O2 ⟨ R13 - Lack of robustness of	

Risk Balance Profile
Software Quality and V&V Program Guide
"FODORS"

Program	Minimal QA/V&V Program (Very High Risk)	Tailored Approach	Complete QA/V&V Program (Low Risk)
Program Content RESIDUAL PERFORMANCE RISK	**Program Content** **Testing** T1-Accept Test (pass/fail w/o metrics) T2-Functional Test (pass/fail) **Analysis** A1-Hazards Analysis A2-S/W FMEA (if applicable for critical functions only) **QA** None **Related Management** None **Other** None **Residual Risks (Missing Content)** 1- Lack of confidence in acceptability of S/W to meet system's needs-T1+ 2 - Unknown functional and system margins-T2+ 3 - Inconsistent S/W requirements with respect to the system's functional requirements (FRD)-Q2 4 - Incorrect design functionality-Q2 5 - No regression testing -T5, M4 6 - S/W builds not converging to an acceptable product – T5, M2 7 - Inputs to S/ W could violate boundary conditions, trigger non-tested paths, etc – T5,Q2 8 - Poor Workmanship in the software product (spaghetti code, un-maintainable code, etc.) – Q1, Q3 9 - Latent S/W defects could cause the system to fail or not meet it's requirements- T5, Q2, Q5 10 - Late awareness (or lack of anticipation) of schedule, performance, cost and quality problems – T5, Q5, M2, M3 11 - Software safety problem – A2+, A3 12 - Executing faulty commands on a spacecraft – Q1, O2 13 - Lack of robustness of functions supported by S/W – Q3, Q5, A4 14 - S/W fails in a harmful manner – A1+, A2+ 15 - H/W and system failures compounded by inappropriate S/W responses – Q5, M4 16 - Missing, wrong or extra software requirements -Q2, Q3, Q5, M2 17 - Working with out of date requirements - Q2, Q3, Q5, M2 18 - Failure to identify critical QA and V&V processes for S/W – T1+, T2+, A2+ 19 - Failure to identify critical contractor monitor points – M2, M3, O1 20 – Can't identify changes impacts (cost, schedule, functionality, etc.) – M2 21 - Project progressing to the next phase of development before it is ready – M1, M3 22 - Non-standard documentation and source code - M1, M2 23 - Unable to effectively add personnel to an "in progress" project – T5, M1 24 - Unable to make enhancements and changes to the S/W – Q1, Q2, M2 25 - Un-reusable S/W products – Q2 26 - Choosing the wrong/high risk contractor to develop software – M4, O1 27 - Receiving wrong RFP responses with respect to S/W – M1, M2 28 - Encountering a S/W error that wasn't tested – M1, M4 29 - Uploading faulty software to a spacecraft after launch – T5, M2, O2	**Program Content** *As Selected (Tailored to be Project Specific)* **Residual Risks** *Appropriate Subset of Residual Risk Issue Relating to Selected Program Content*	**Program Content** **Testing** T1-Accept Test (w/ Metrics, full functional coverage, & witnessing) T2-Full Functional Test (w/ Metrics) T3-Subsystem integration Test (Metrics / trend analysis) T4-Unit Test full SW Dev Folders) T5-Formal Test Plan **Analysis** A1-Hazards Analysis A2-S/W FMEA A3-Safety Analysis (Full) A4-Code Analysis (Full) A5-S/W Fault Tree Analysis **QA** Q1-Conformance to S/W Standards & Guidelines (QA critical item audit) Q2-Requirements Trace (complete) Q3-S/W Inspections(NASA) used for TSRs (*w/ increased coverage*). *Q4-IV&V w/independent reporting to NASA HQ* Q5-Requirements Mgt. (trace CM, CCB,. tool, . *volatility tracking)* Q6-Operations Software QA & V&V (incremental updates) **Related Management** M1-Full S/W QA Plan M2-Configuration Management *(Full coverage w/ mandatory use of a tool)* M3-Milestone Reviews (CDR, PDR, etc. *with participation of independent reviewers mandatory)* M4-Project Risk Management program M5-Integrated Support of Fault Protection and/or Failure Detection, Isolation & Recovery subsystems M6-Full Project Software Metrics program **Other** O1-Support Contractor Mgt. (continuous assessment w/ RFP & SEB support from QA *and IV&V* roles) O2-Mission Operations and Command Assurance (MOCA) **Residual Risks (Missing Content)** 28 - Encountering a S/W error that wasn't tested (i.e., can't test everything in a complex software product) – M1+, M4+ 29 - Uploading faulty software to a spacecraft after launch – T5+, M2+, O2+

	Mitigations (Risk Reduction)	Mitigations	Mitigations (Risk Reduction)
MITIGATIONS	1 - Use of an Automatic Code Generator(8,22,24) 2 - Reusing high quality proven software products (req., design, code, and/or test cases)(1,7,8,9,13,22,25) 3 - Using Rapid Prototyping aspects of the software system(1,3,6,16) 4 - Simulation of software subsystem(1,3,10,12,16,24,29) 5 - Embedding Assertions in the code (1,3,14,16,17) 6 - Lessons learned(1,5,10,18,19,20,21,22,26) 7 - Apply PACTS to critical functions(1,3,6,9,10,29) 8 - Identify critical functions(1,3,10,13,16,17,21,27,28) 9 - Establish volatility metrics(1,5,9,21,24,28) 10 - Use Complexity metrics(1,4,7,8,28) 11 - Early training(8,9,16,22,23,24,29) 12 - Cross training(8,9,16,22,23,24,29) 13 - Do regression testing(1,3,5,9,14,24,28) 14 - Incentivize contractor(5,8,9,10,21,24,26) 15 - Establish reuse requirements(1,6,8,9,13,20,22,24,,25,28) 16 - Use TSRs (incl auto code gen)(1,4,8,10,16,17,20,21,22,25,28) 17 - Insight review of contractor SEI level(8,10,19,21,22,26) 18 - Use EVA metrics(10,20,26) 19 - Standard documentation formats, reports(6,8,10,16,20,21,22,23,24,25) 20 - Validation of auto code generator(5,6,7,8,9) 21 - Augmenting V&V with Formal Methods techniques (1,3,14,16,17, 28)		2 - Reusing high quality proven software products (req., design, code, and/or test cases) (*) 6 - Lessons learned (*) 7 - Apply PACTS to critical functions (29) 12 - Cross training (29) 14 - Incentivize contractor (*) 15 - Establish reuse requirements (28) 21 - Augmenting traditional V&V with Formal Methods techniques (formal specification, model checking, animating specifications, *and/or proofs*)) (28, *) Note: * indicates a general risk reduction

Note: + indicates that adding stronger content techniques of type could reduce this risk